

Claims

We claim:

- Sub A3 7
1. A method of creating a graphical program to perform an algorithm, the method comprising:
 - recording one or more functions in response to user input, wherein the one or more functions specify the algorithm; and
 - automatically generating the graphical program in response to the recorded one or more functions, wherein the graphical program implements the algorithm.
 2. The method of claim 1, further comprising:
 - performing the one or more functions in response to user input;
 - wherein said recording the one or more functions is performed in response to said performing the one or more functions.
 3. The method of claim 1,
 - wherein said recording the one or more functions comprises creating a prototype.
 4. The method of claim 3,
 - wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:
 - image processing, machine vision, image analysis, robotics, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.
 5. The method of claim 1,
 - wherein said recording the one or more functions is performed in response to user input received via a graphical user interface (GUI).
 6. The method of claim 5,
 - wherein the graphical user interface is associated with a prototyping environment application.

7. The method of claim 5,
wherein the user input comprises selecting the functions from one or more of a
menu or palette.

8. The method of claim 1,
wherein said automatically generating the graphical program comprises
programmatically generating the graphical program in response to the recorded one or
more functions.

9. The method of claim 1,
wherein said automatically generating the graphical program comprises
automatically generating graphical code in the graphical program without direct user
input.

10. The method of claim 1, further comprising:
executing the graphical program to perform the algorithm.

11. The method of claim 1,
wherein the graphical program includes a block diagram portion and a user
interface panel portion.

12. The method of claim 6,
wherein the graphical program is a graphical data flow program.

13. The method of claim 1,
wherein said automatically generating the graphical program comprises including
one or more nodes corresponding to the one or more functions in the graphical program.

14. The method of claim 1, wherein the recorded one or more functions
comprise a script, the method further comprising:

creating an association between the script and the graphical program;
modifying the script to create a new script in response to user input after said
creating the association; and
modifying the graphical program according to the new script to create a new
5 graphical program.

15. The method of claim 14,
wherein said modifying the graphical program according to the new script uses
the association between the script and the graphical program;
10 wherein the association remains between the new script and the new graphical
program.

16. The method of claim 14, further comprising:
receiving user input indicating a desire to change the graphical program;
15 displaying script information of the script;
modifying the script information in response to user input; and
modifying the graphical program after said modifying the script information.

17. The method of claim 1, further comprising:
20 creating an association between the script and the graphical program;
locking the association between the script and the graphical program, wherein
said locking prevents user editing of the graphical program.

18. The method of claim 17, further comprising:
25 unlocking the association between the script and the graphical program in
response to user input after said locking;
directly changing the graphical program in response to user input after said
unlocking.

30 19. The method of claim 18,

wherein said unlocking removes the association between the script and the graphical program.

20. The method of claim 17, further comprising:

5 modifying the graphical program in response to user input after said generating the graphical program and after said creating the association between the script and the graphical program;

determining if an association exists between the script and the graphical program in response to said modifying the graphical program; and

10 removing the association between the script and the graphical program in response to said modifying.

21. The method of claim 1, further comprising:

receiving user input specifying code generation information;

15 wherein said automatically generating the graphical program utilizes the code generation information.

22. The method of claim 21,

20 wherein the code generation information specifies a type of graphical program to create in response to the recorded one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

23. The method of claim 22,

25 wherein the graphical program type specifies a particular graphical programming environment;

wherein the graphical program is created in a file format that is usable by the particular graphical programming environment.

30 24. The method of claim 21,

wherein a plurality of parameters are associated with the one or more functions,
wherein each parameter is an input parameter which provides input to a function or an
output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input
5 parameters which are desired to be interactively changeable or one or more of the output
parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling
the graphical program to receive user input during program operation, wherein the user
input specifies values for the specified one or more input parameters;

10 wherein said automatically generating the graphical program comprises enabling
the graphical program to display output during program operation, wherein the output
indicates values for the specified one or more output parameters.

25. The method of claim 1,
15 wherein said automatically generating the graphical program comprises:
generating portions of graphical code, wherein each portion of graphical
code implements one of the functions;
linking the portions of graphical code together.

20 26. The method of claim 25,
wherein each portion of graphical code includes one or more graphical program
nodes, wherein each node has one or more inputs or outputs;
wherein generating each portion of graphical code comprises connecting the node
inputs and outputs together in order to implement the function with which the portion of
25 graphical code is associated.

27. The method of claim 26,
wherein linking a first portion of graphical code to a second portion of graphical
code comprises connecting an output of a node in the first portion of graphical code to an
30 input of a node in the second portion of graphical code.

28. The method of claim 26,
wherein at least one of the functions has an associated input parameter;
wherein each portion of code that implements a function that has an associated
input parameter includes a node that has an input for receiving a value for the input
parameter;

wherein each portion of code that implements a function that has an associated
input parameter includes a leaf node that has an output for providing a value for the input
parameter;

wherein the leaf node output for providing the parameter value is connected to the
node input for receiving the parameter value.

29. The method of claim 26,
wherein at least one of the functions has an associated output parameter;
wherein each portion of code that implements a function that has an associated
output parameter includes a node that has an output for providing a value for the output
parameter;

wherein each portion of code that implements a function that has an associated
output parameter includes a leaf node that has an input for receiving a value for the
output parameter;

wherein the leaf node input for receiving the parameter value is connected to the
node output for providing the parameter value.

30. The method of claim 25, further comprising:
for each function, retrieving information associated with the function from a
database;

wherein generating the portion of graphical code that implements a particular
function utilizes the database information retrieved for the particular function.

31. A system for creating a graphical program to perform an algorithm, the
system comprising:
a processor;

a memory coupled to the processor which stores a prototyping environment application,

a user input device which receives user input;

wherein the prototyping environment application is executable in response to the user input to store one or more functions in the memory, wherein the one or more functions specify the algorithm;

wherein the prototyping environment application is executable to automatically generate a graphical program in response to the stored one or more functions, wherein the graphical program implements the algorithm specified by the one or more functions.

32. The system of claim 31,

wherein said storing the one or more functions in the memory comprises creating a prototype.

33. The system of claim 32,

wherein the prototyping environment application is a prototyping environment application in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, robotics, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.

34. The system of claim 31,

wherein the prototyping environment application includes a graphical user interface (GUI);

wherein said storing the one or more functions in the memory is performed in response to user input received via the graphical user interface.

35. The system of claim 34,

wherein the user input comprises selecting the functions from one or more of a menu or palette.

36. The system of claim 31, further comprising:

a graphical program creation program stored in the memory;

wherein the prototyping environment application is executable to call the graphical program creation program;

wherein the graphical program creation program is executable to automatically
5 generate the graphical program in response to said prototyping environment application
calling the graphical program creation program.

37. The system of claim 36,

wherein the graphical program creation program is a graphical programming
10 development environment application.

38. The system of claim 31,

wherein said automatically generating the graphical program comprises
programmatically generating the graphical program in response to the stored one or more
15 functions.

39. The system of claim 31,

wherein said automatically generating the graphical program comprises
automatically generating graphical code in the graphical program without direct user
20 input.

40. The system of claim 31,

wherein the graphical program includes a block diagram portion and a user
interface portion.

41. The system of claim 39,

wherein the graphical program is a graphical data flow program.

42. The system of claim 31,

wherein said automatically generating the graphical program comprises including
30 one or more nodes corresponding to the one or more functions in the graphical program.

43. The system of claim 36,
wherein the stored one or more functions comprise a script;
wherein the memory stores an association between the script and the graphical
5 program;

wherein the prototyping environment application is executable to utilize the
association to modify the script to create a new script in response to user input;

wherein the graphical program creation program is executable to modify the
graphical program according to the new script to create a new graphical program.

44. The system of claim 43,
wherein the memory stores an association between the script and the graphical
program;

wherein the memory stores information specifying that the association between
15 the script and the graphical program is locked, wherein said locking prevents user editing
of the graphical program.

45. The system of claim 31,
wherein the prototyping environment application is executable to receive user
20 input specifying code generation information, wherein the code generation specifies
information to use in generating the graphical program.

46. The system of claim 45,
wherein the code generation information specifies a type of graphical program to
25 create in response to the stored one or more functions;
wherein the graphical program is created in accordance with the specified
graphical program type.

47. The method of claim 46,
30 wherein the graphical program type specifies a particular graphical programming
environment;

wherein the graphical program is created in a file format that is usable by the particular graphical programming environment.

48. The system of claim 45,

wherein a plurality of parameters are associated with the functions, wherein each parameter is an input parameter which provides input to a function or an output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input parameters which are desired to be interactively changeable or one or more of the output parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input parameters;

wherein said automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified one or more output parameters.

49. The system of claim 31,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

50. The system of claim 49,

wherein each portion of graphical code includes one or more graphical program nodes, wherein each node has one or more inputs or outputs;

wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated.

51. The system of claim 50,

wherein linking a first portion of graphical code to a second portion of graphical code comprises connecting an output of a node in the first portion of graphical code to an input of a node in the second portion of graphical code.

5 52. The system of claim 49,
 wherein, for each function, information associated with the function is retrieved from a database;
 wherein generating the portion of graphical code that implements a particular function utilizes the database information retrieved for the particular function.

10 53. A memory medium comprising program instructions executable to:
 record one or more functions in response to user input, wherein the one or more functions specify an algorithm; and
 automatically generate a graphical program in response to the recorded one or
15 more functions, wherein the graphical program implements the algorithm.

 54. The memory medium of claim 53, further comprising program instructions executable to:
 perform the one or more functions in response to user input;
20 wherein said recording the one or more functions is performed in response to said performing the one or more functions.

 55. The memory medium of claim 53,
 wherein said recording the one or more functions comprises creating a prototype.

25 56. The memory medium of claim 55,
 wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:
 image processing, machine vision, image analysis, robotics, process control,
30 industrial automation, test and measurement, simulation, workflow processes, and robotics.

57. The memory medium of claim 53,
wherein said recording the one or more functions is performed in response to user
input received via a graphical user interface (GUI).

5 58. The memory medium of claim 57,
wherein the graphical user interface is associated with a prototyping environment
application.

10 59. The memory medium of claim 57,
wherein the user input comprises selecting the functions from one or more of a
menu or palette.

15 60. The memory medium of claim 53,
wherein said automatically generating the graphical program comprises
programmatically generating the graphical program in response to the recorded one or
more functions.

20 61. The memory medium of claim 53,
wherein said automatically generating the graphical program comprises
automatically generating graphical code in the graphical program without direct user
input.

25 62. The memory medium of claim 53, further comprising program instructions
executable to:
execute the graphical program to perform the algorithm.

30 63. The memory medium of claim 53,
wherein the graphical program includes a block diagram portion and a user
interface panel portion.

64. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises including one or more nodes corresponding to the one or more functions in the graphical program.

65. The memory medium of claim 53, wherein the recorded one or more functions comprise a script, the memory medium further comprising program instructions executable to:

create an association between the script and the graphical program;

modify the script to create a new script in response to user input after said creating the association; and

modify the graphical program according to the new script to create a new graphical program.

66. The memory medium of claim 65, further comprising program instructions executable to:

create an association between the script and the graphical program;

lock the association between the script and the graphical program, wherein said locking prevents user editing of the graphical program.

67. The memory medium of claim 53, further comprising program instructions executable to:

receive user input specifying code generation information;

wherein said automatically generating the graphical program utilizes the code generation information.

68. The memory medium of claim 67, wherein the code generation information specifies a type of graphical program to create in response to the recorded one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

69. The memory medium of claim 67,

wherein a plurality of parameters are associated with the one or more functions,
wherein each parameter is an input parameter which provides input to a function or an
output parameter which accepts output from a function;

5 wherein the code generation information specifies one or more of the input
parameters which are desired to be interactively changeable or one or more of the output
parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling
the graphical program to receive user input during program operation, wherein the user
input specifies values for the specified one or more input parameters;

10 wherein said automatically generating the graphical program comprises enabling
the graphical program to display output during program operation, wherein the output
indicates values for the specified one or more output parameters.

70. The memory medium of claim 53,

15 wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical
code implements one of the functions;

linking the portions of graphical code together.

20 71. A method of creating a graphical program to perform an algorithm, the
method comprising:

creating a prototype in response to user input, wherein the prototype specifies the
algorithm; and

25 automatically generating the graphical program in response to the prototype,
wherein the graphical program implements the algorithm.

72. The method of claim 71,

wherein the prototype comprises a prototype in at least one of the disciplines from
the group consisting of:

image processing, machine vision, image analysis, robotics, process control, industrial automation, test and measurement, simulation, telecommunications, workflow processes, and robotics.

5 73. The method of claim 71,
 wherein the user input is received via a graphical user interface (GUI) associated
with a prototyping environment application.

10 74. The method of claim 73,
 wherein the user input comprises selecting one or more functions from one or
more of a menu or palette.

15 75. The method of claim 71,
 wherein said automatically generating the graphical program comprises
programmatically generating the graphical program in response to the created prototype.

20 76. The method of claim 71,
 wherein said automatically generating the graphical program comprises
programmatically generating graphical code in the graphical program without direct user
input.

25 77. The method of claim 71,
 wherein said automatically generating the graphical program comprises including
one or more function nodes in the graphical program.

 78. The method of claim 71,
 wherein said creating the prototype in response to user input comprises creating a
diagrammatic model of the algorithm.

30 79. The method of claim 71,

9. The method of claim 79,
herein said automatically generating portions of graphs;
generating portions of graphs;
elements one of the functions;
linking the portions of graphs

10

	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---